Alex L. Demidov

Tbilisi, Georgia | +995 511 144 213 | alexeydemidov@gmail.com
linkedin.com/in/alekseydemidov | github.com/AlexeyDemidov

Technical Skills

With 20+ years of experience can single-handedly build a well-architected, secure, high-performing, resilient and cost-effective infrastructure for a small/medium scale web application with all necessary components: fully automated infrastructure as code, container orchestration, monitoring and observability, automated testing and deployment pipelines, source code repository management etc.

Container and infrastructure tools: AWS, Kubernetes, Nomad, Docker, podman, Hashicorp Consul, Vault, Packer
Configuration Management and IaC: Terraform, Ansible, Chef, Puppet
Observability: Grafana, Loki, Prometheus, ELK stack, CloudWatch
CI/CD: Jenkins, GitHub Actions, Flux, GitLab
Languages: Ruby, Python, bash, C

Experience

Lead DevOps/SRE Engineer Dec 2011 – Present
rebuildingsociety.com Ltd. *Leeds, UK*

- Responsible for the entire platform infrastructure.
- Inherited the project as a simple LAMP stack web application deployed to a single dedicated server with FTP. To get the project into a manageable state, moved the application code to a self-hosted gitolite repository, implemented automated deployment on 'git push' and isolated the application into a separate environment in LXC (Docker didn't exist at that time) managed with chef-solo (Ansible didn't exist either).
- As the project quickly grew, split it into multiple LXC containers on geographically separated dedicated servers (connected with IPsec tunnels) for high availability with a distributed replicated MySQL database with automatic fail-over, distributed Redis for caching and a shared GlusterFS file system. Created separate isolated dev and staging environments. For better availability, added load balancers. Integrated NewRelic for performance monitoring and tuning. Added Jenkins and Selenium to run integration tests.
- Later, migrated the primary application, the database and the load balancers to AWS for elasticity and geographic distribution. The AWS infrastructure is managed with Terraform and tested with serverspec, AMI images are generated by packer.
- To better utilise available capacity, containerized some parts and they are now running using Nomad container orchestrator which allows running containers both on AWS and the legacy dedicated servers to save costs. Jenkins is used to build docker images and push them to AWS ECR and local JFrog Container Registry.
- The services are constantly monitored and run with 99.98% availability. The infrastructure was able to sustain a targeted DDoS attack. As a result of proper use of caching and CDN, the infrastructure survived a Slashdot effect with a 200 times traffic increase with only 2 times cost increase. AWS infrastructure costs are constantly monitored using a custom report with daily cost changes per service.
- Security is constantly monitored and enforced. Daily security software updates, and daily security infrastructure scans from external locations. AWS infrastructure is monitored with AWS Config, GuardDuty and Inspector. Sensitive information is encrypted and passwords and TLS certificates are stored in Hashicorp Vault and AWS KMS. TLS certificates are automatically checked daily for expiration.
- Created a threat model using STRIDE methodology. Performed risk analysis and gap analysis. Improved the company's security policies. Improved vulnerability handling of 3rd party components and automated CVE discovery and reporting. Lead developers in fixing or replacing vulnerable 3rd party components.
- All infrastructure is managed as code with Terraform, Ansible and Chef. Terraform drift and the server configuration drift are checked daily automatically. Base AMIs are rebuilt regularly. Cloud instances are periodically destroyed and rebuilt.
- Migrated a separate project infrastructure from a number of dedicated instances (multiple frontends, backends and DB) on Azure into containers running on nomad cluster with a separate Traefik ingress. Added missing monitoring/observability tooling (ELK stack). Created regular backups. Added CI/CD pipeline using Jenkins and JFrog Container Registry. The resulting cost saving is close to 100 times.

SRE/DevOps Consultant Nov 2019 – Present

Undisclosed Client *UK*

- Migrated a Ruby on Rails app to a self-hosted Kubernetes cluster with a HAProxy ingress. In the same cluster deployed PostgreSQL, ELK cluster stack for log collection, Prometheus for monitoring and alerting, Jenkins CI/CD pipeline, Flux CD, JFrog Container Registry, later migrated to Harbor. Ongoing migration to GitLab.

SRE/DevOps Consultant Nov 2021 – Dec 2021

Undisclosed Client *US*

- Built a test self-contained setup with PostgreSQL async replication with artificially induced configurable lag between primary and replica servers to allow the client to debug read-after-write consistency issues in his application.

Lead Ruby on Rails Developer Oct 2011 – Present

nouri.sh *Leeds, UK*

- Besides writing the code, managed the project infrastructure, including a self-hosted email server to handle mass email delivery for RSS-to-email service. In-depth knowledge of email delivery and appropriate RFCs. Migrated the email server from qmail to postfix to properly support DKIM, DMARC and TLS

Lead System Administrator Feb 1999 – Mar 2009

Variant-Inform *Dimitrovgrad, Russia*

- As ISP lead engineer, was responsible for the implementation and maintenance of a wide range of services: Backbone network and WAN links, Cisco routers and switches, xDSL modems, OSPF, Dial-up Access Servers (Cisco AS, Lucent Portmaster), PPPoE access servers. Network security - packet filtering, IDS snort. Network monitoring - SNMP, nagios, cacti, mon. Backups. Unix systems - FreeBSD/Linux. Network services: NIS, NFS, DNS, Tacacs, FreeRadius. Mail services, sendmail/postfix/clamav/drweb/cyrus. Web services, apache 1.x/2.x, mod_perl, mod_php. SQL servers PostgreSQL, MySQL
- In 2000-2002 led in-house development of ISP billing system integrated with automatic configuration management. The system was developed with Perl/CGI and used PostgreSQL backend. The system remained in production use well into the 2010s.
- In 2005 led another in-house development of a billing system for a cable TV provider company. This system was developed with mod_perl2, Template Toolkit, Class::DBI, XML/XSLT templates, PostgreSQL.

Education

Moscow Engineering Physics Institute Moscow

Automated Information Processing and Control Systems

Notes

I'm occasionally active on serverfault.com and related sites. Example answers about Linux OOMs or cgroup memory limits or DNS terminology or capacity estimation or web-server performance troubleshooting or Docker troubleshooting and my own troubleshooting question and security question should demonstrate my approach to troubleshooting and communication.

Blog posts

Falsehoods People Believe About Linux Swap and OOM

Linux Active and Inactive Memory: The Deceptive Labels